

This listing of claims will replace all prior versions, and listings, of claims in the application:

The Status of the Claims

1. (Currently Amended) A method of prefetching memory objects comprising:
 - identifying program states associated with an executing program;
 - associating memory profiles with respective ones of the program states;
 - identifying at least one next probable state; ~~and~~
 - prefetching memory objects associated with the at least one memory profile corresponding to the at least one next probable state; and
 - determining entropy values associated with the identified program states, wherein identifying the at least one next probable state comprises identifying the at least one next probable state based on the entropy values.
2. (Previously Presented) A method as defined in claim 1 further comprising updating at least one of the memory profiles.
3. (Previously Presented) A method as defined in claim 2 wherein updating at least one of the memory profiles comprises filtering the at least one of the memory profiles.
4. (Previously Presented) A method as defined in claim 3 wherein filtering the at least one of the memory profiles comprises removing at least one

memory reference from the at least one of the memory profiles.

5. (Previously Presented) A method as defined in claim 3 wherein filtering the at least one of the memory profiles comprises miss filtering the at least one of the memory profiles.

6. (Previously Presented) A method as defined in claim 3 wherein filtering the at least one of the memory profiles comprises usage filtering the at least one of the memory profiles.

7. (Previously Presented) A method as defined in claim 1 further comprising developing a trace of the executing program.

8. (Previously Presented) A method as defined in claim 7 wherein developing the trace comprises developing at least one of a program counter trace, an instruction pointer trace, a basic block trace, and a memory address trace.

9. (Previously Presented) A method as defined in claim 7 wherein identifying the program states comprises:

assigning possible state signatures to sets of entries in the trace;

selecting one of the possible state signatures as a first state signature;

comparing the first state signature to at least one subsequent one of the possible state signatures; and

if the at least one subsequent one of the possible state signatures differs

from the first state signature by at least a predetermined amount, identifying the subsequent one of the possible state signatures as a second state signature.

10. (Previously Presented) A method as defined in claim 9 wherein the predetermined amount is selected such that the program states have a duration which is greater than a latency associated with a memory object and less than a time which causes cache pollution.

11. (Previously Presented) A method as defined in claim 9 wherein assigning possible state signatures to sets of entries in the trace comprises:

developing a first possible state signature from a first set of entries in the trace; and

developing a second possible state signature from a second set of entries in the trace, the first set of entries partially intersecting the second set of entries.

12. (Previously Presented) A method as defined in claim 11 wherein developing the first possible state signature for the first set of entries in the trace comprises:

weighting members of the first set such that later members have greater weight than earlier members; and

mapping the weighted members to a bit vector signature.

13. (Cancelled)

14. Cancelled)

15. (Currently Amended) A method as defined in claim ~~13~~ 1 wherein determining the entropy values associated with the identified program states comprises:

determining probabilities of transitioning from a first one of the program states to a plurality of the program states; and

converting the probabilities to an entropy value for the first program state.

16. (Previously Presented) A method as defined in claim 15 wherein the probabilities comprise at least one of moving averages and exponential moving averages.

17. (Currently Amended) A method as defined in claim ~~13~~ 1 wherein determining the entropy values comprises associating next state probability profiles with respective ones of the program states.

18. (Currently Amended) A method as defined in claim ~~13~~ 17 further comprising updating at least one of the next state probability profiles.

19. (Previously Presented) A method as defined in claim 1 wherein the at least one next probable state comprises a next most probable state.

20. (Previously Presented) A method as defined in claim 1 wherein the at least one next probable state comprises at least two next most probable states.

21. (Currently Amended) A method as defined in claim ~~13~~ 1 wherein if the entropy value of a current program state is less than a first predetermined threshold, prefetching the memory objects associated with the at least one memory profile corresponding to the at least one next probable state comprises prefetching the memory objects associated with a next most probable state.

22. (Previously Presented) A method as defined in claim 21 wherein if the entropy value of a current program state is greater than a second predetermined threshold, prefetching the memory objects associated with the at least one memory profile corresponding to the at least one next probable state comprises prefetching the memory objects associated with a plurality of next probable states.

23. (Previously Presented) A method as defined in claim 22 wherein the first threshold and the second threshold are substantially identical.

24. (Previously Presented) A method as defined in claim 21 wherein prefetching the memory objects is not performed if the entropy value of a current program state is greater than a second predetermined threshold.

25. (Previously Presented) A method as defined in claim 24 wherein the first threshold and the second threshold are substantially identical.

26. (Previously Presented) A method as defined in claim 1 wherein prefetching the memory objects is performed near a beginning of a current program state.

27. (Previously Presented) A method as defined in claim 1 wherein identifying program states comprises defining the program states to have a duration which is greater than a latency associated with a memory operation and less than a time which causes cache pollution.

28. (Currently Amended) An article of manufacture storing machine readable instruction which, when executed, cause a machine to:

identify program states associated with an executing program;

associate memory profiles with respective ones of the program states;

identify at least one next probable state; ~~and~~

prefetch memory objects associated with the at least one memory profile corresponding to the at least one next probable state; and

determine entropy values associated with the identified program states,
wherein the machine readable instructions cause the machine to identify the at
least one next probable state based on the entropy values.

29. (Previously Presented) An article of manufacture as defined in claim 28 wherein the machine readable instructions further cause the machine

to update at least one of the memory profiles.

30. (Previously Presented) An article of manufacture as defined in claim 29 wherein the machine readable instructions further cause the machine to update the at least one of the memory profiles by filtering the at least one of the memory profiles.

31. (Previously Presented) An article of manufacture as defined in claim 30 wherein filtering the at least one of the memory profiles comprises removing at least one memory reference from the at least one of the memory profiles.

32. (Previously Presented) An article of manufacture as defined in claim 30 wherein filtering the at least one of the memory profiles comprises miss filtering the at least one of the memory profiles.

33. (Previously Presented) An article of manufacture as defined in claim 30 wherein filtering the at least one of the memory profiles comprises usage filtering the at least one of the memory profiles.

34. (Cancelled)

35. (Cancelled)

36. (Previously Presented) An article of manufacture as defined in claim 28 wherein the at least one next probable state comprises a next most probable state.

37. (Previously Presented) An article of manufacture as defined in claim 28 wherein the at least one next probable state comprises at least two next most probable states.

38. (Currently Amended) An article of manufacture as defined in claim ~~34~~ 28 wherein if the entropy value of a current program state is less than a first predetermined threshold, the machine readable instructions cause the machine to prefetch the memory objects associated with the at least one memory profile corresponding to a next most probable state.

39. (Previously Presented) An article of manufacture as defined in claim 38 wherein if the entropy value of a current program state is greater than a second predetermined threshold, the machine readable instructions cause the machine to prefetch the memory objects associated with the at least one memory profile corresponding to a plurality of next probable states.

40. (Previously Presented) An article of manufacture as defined in claim 38 wherein the machine readable instructions cause the machine to not prefetch memory objects if the entropy value of a current program state is greater than a second predetermined threshold.

41. (Currently Amended) An apparatus to prefetch memory objects comprising:

a state identifier to identify program states;

a predictor to identifying at least one next probable state, wherein the predictor determines entropy values associated with the program states identified by the state identifier; and

a prefetcher to prefetch at least one memory object associated with the at least one next probable state based on the entropy values.

42. (Previously Presented) An apparatus as defined in claim 41 wherein the at least one next probable state comprises a next most probable state.

43. (Previously Presented) An apparatus as defined in claim 41 wherein the at least one next probable state comprises at least two next most probable states.

44. (Previously Presented) An apparatus as defined in claim 41 further comprising a memory state monitor to associate memory profiles with respective ones of the program states.

45. (Previously Presented) An apparatus as defined in claim 44 wherein the at least one memory object prefetched by the prefetcher is identified in a memory profile associated with the at least one next probable

state.

46. (Previously Presented) An apparatus as defined in claim 44
wherein the memory state monitor updates at least one of the memory profiles.

47. (Previously Presented) An apparatus as defined in claim 46
wherein the memory state monitor updates the at least one of the memory
profiles by filtering the at least one of the memory profiles.

48. (Previously Presented) An apparatus as defined in claim 47
wherein the memory state monitor filters the at least one of the memory
profiles by removing at least one memory reference from the at least one of the
memory profiles.

49. (Previously Presented) An apparatus as defined in claim 47
wherein the memory state monitor filters the at least one of the memory
profiles by miss filtering the at least one of the memory profiles.

50. (Previously Presented) An apparatus as defined in claim 47
wherein the memory state monitor filters the at least one of the memory
profiles by usage filtering the at least one of the memory profiles.

51. (Previously Presented) An apparatus as defined in claim 41
further comprising a trace sampler to develop a trace of a program.

52. (Previously Presented) An apparatus as defined in claim 51 wherein the state identifier further comprises:

 a signature developer to develop possible state signatures for sets of entries in the trace;

 a state distinguisher to identify program states based on the possible state signatures; and

 a memory to store the state signatures of the program states identified by the state distinguisher.

53. (Previously Presented) An apparatus as defined in claim 52 wherein the memory stores a data structure including at least one of: a signature of a first program state, an age of the first program state, a usage frequency of the first program state, an entropy value of the first state, a set of probabilities of transitioning from the first program state to a set of program states, and a set of memory references associated with the first program state.

54. (Cancelled)

55. (Currently Amended) An apparatus as defined in claim ~~54~~ 1 wherein if the entropy value of a current program state is less than a first predetermined threshold, the at least one memory object prefetched by the prefetcher is associated with a next most probable state.

56. (Previously Presented) An apparatus as defined in claim 55 wherein if the entropy value of a current program state is greater than a second

predetermined threshold, the at least one memory object prefetched by the prefetcher is associated with a plurality of next probable states.

57. (Previously Presented) An apparatus as defined in claim 55 wherein if the entropy value of a current program state is greater than a second predetermined threshold, the prefetcher does not prefetch the at least one memory object..

58. (Previously Presented) An apparatus as defined in claim 41 wherein the prefetcher prefetches the memory objects near a beginning of a current program state.

59. (Previously Presented) An apparatus as defined in claim 41 wherein the state identifier defines the program states to have a duration which is greater than a latency associated with a memory operation and less than a time which causes cache pollution.

60. (Previously Presented) An apparatus as defined in claim 41 wherein the predictor further comprises:
a state transition monitor to calculate probabilities of transitioning from a first one of the program states to a plurality of program states; and
an entropy calculator to convert the probabilities to an entropy value for the first program state.

61. (Previously Presented) An apparatus as defined in claim 41 wherein the predictor further comprises an event predictor to predict at least one next probable program state.

62. (Cancelled)

63. ((Cancelled))

64. (Cancelled)